

## Projektityön dokumentti – Älykäs reseptikirja

*Korjattu versio*

### Yleiskuvaus

Älykäs reseptikirja on ohjelma, joka ylläpitää ruokareseptikirjaa, jonka avulla voi helposti etsiä kriteerit täyttävä ruokalaji. Käyttäjällä on ajanmukaiset tiedot ruoka-aineista, jotka ovat sillä hetkellä varastossa. Ohjelma voi etsiä ruokalajeja, jotka voidaan valmistaa käymättä kaupassa, tai ruokalajit, jotka vaativat N puuttuvaa tai osittain puuttuvaa ainetta. Reseptejä voidaan hakea myös niiden sisältämien ruoka-aineiden perusteella.

Aineisiin voidaan liittää merkintä niiden sisältämistä allergeeneistä, jotta voidaan rajoittaa hakua, jos halutaan välttää esimerkiksi maitoa sisältäviä ruokalajeja. Aineet voidaan myös rakentaa itse reseptistä samoin kuin ruokalajit, esimerkiksi jauhelihapihvin resepti voi sisältää lihamureketaikinaa, jolla on puolestaan oma reseptinsä. Jos lihamureketaikina puuttuu, ohjelma yrittää koota sen raaka-aineistaan. Ohjelma käsittelee sekä reseptejä että ruoka-aineita ”aineina”: reseptit ovat aineita, joita voidaan valmistaa muista aineista, joita kutsutaan ohjelmassa ainesosiksi. Ainesosat, joilla ei ole omia ainesosiaan, kutsutaan raaka-aineiksi.

Ohjelmassa on myös luokka, Muuntaja, joka tekee yksikkömuunnokset aineille niiden tiheyttä käyttäen. Ohjelmassa kaikkien aineiden tiheyden mittayksikkö on g/ml. Muuntaja on olennainen, koska resepteissä aineet voivat ilmetä eri mittayksikössä kuin tavallisesti. Esimerkiksi jauhoja voidaan ostaa kiloittain, mutta reseptit voivat mitata niitä desilitroina.

Ohjelmassa on graafinen käyttöliittymä, jolla voi käyttää kaikkia ohjelman toimintoja, kuten hallita varastoa ja luoda uusia aineita. Ohjelma on siis pyritty toteuttamaan keskivaikealla tasolla.

### Käyttöohje

Ohjelma käynnistetään GUI-objektin kautta. Siitä aukeaa ohjelman pääikkuna, jossa on tekstitaulukko, jossa on listattuna kaikki ohjelman tuntemat aineet sekä niiden määrät ja allergeenit. Ikkunassa on myös kuusi nappia: Reseptihaku, Luo Resepti, Varastonhallinta, Avaa aine, Tiheyslaskuri ja Sulje Ohjelma. Nämä avaavat ikkunoita, joissa voidaan suorittaa erinäisiä toimintoja. Reseptihaussa voidaan hakea eri aineita, reseptinluonnissa voidaan luoda uusia aineita ja varaston hallinnassa voidaan muuttaa aineiden määriä ohjelman muistissa. Avaa aine -napilla pääsee tarkastelemaan aineiden ominaisuuksia ja muuttamaan niitä. Tiheyslaskuri helpottaa aineiden luontia laskemalla annetun massan ja tilavuuden avulla niitä vastaavan tiheyden. Sulje ohjelma tallentaa ohjelman tiedot ja sulkee ohjelman.

Aine-olioiden kaikkia muuttujia pystytään muokkaamaan eri metodeilla käyttöliittymän Aine-näkymän kautta. Ikkunasta näkee aineen tiedot ja voi muokata sen ainesosia, allergeeneja ja muita ominaisuuksia sekä selvittää mistä raaka-aineista se koostuu. Ikkunassa on napit näitä toimintoja varten, ja niitä painettua ohjelma pyytää käyttäjältä merkkijonona annetun komennon, jonka se pyrkii toteuttamaan. Komennot ovat tyypillisesti muotoa [operaattori] [ominaisuus] [uusi arvo]. Käyttö on pyritty selittämään kussakin pyynnössä mahdollisimman selkeästi.

Varaston Hallinta -nappi avaa pienen ikkunan, jossa on neljä nappia varaston hallintaan. Nämä ovat aineen määrän ja mittayksikön muuttamiseen, mutta myös yksittäisten tai kaikkien aineiden poistamiseen.

Reseptinluonnissa käyttäjä voi luoda uusia aineita täyttämällä ikkunan tekstikentät, jotka ovat: Aineen nimi, allergeenit, tiheys, määrä ja mittayksikkö, sekä Aineen kuvaus. Näistä ei ole pakko täyttää muita kuin nimi. Nimen voi antaa missä muodossa tahansa, mutta ohjelma muuttaa sen yhtenäiseksi ja pikkukirjaimiseksi merkkijonoksi. Allergeenit erotetaan pilkulla. Tiheys, määrä ja mittayksikkö erotetaan pilkulla, eli annetaan muodossa Double, Double, String. Tiheyden ja määrän tulee olla vähintään 0.0 ja jos ohjelma ei tunnista annettua mittayksikköä, se muutetaan "kpl"-muotoon. Mittayksikön voi muuttaa aineikkunasta. Viimeiseen kenttään voi kirjoittaa aineen kuvauksen.

Ohjelma tunnistaa vain yleisimmät massan ja tilavuuden yksiköt, sekä määritelmän "kpl" joka kuvaa kappaleina tai annoksina mitattavia aineita. Ohjelman käyttämä tiheyden yksikkö on g/ml, jota se käyttää tehdessään yksikkömuunnoksia. Kappalemittaisille aineille ei voi tehdä yksikkömuunnoksia. Tunnetut mittayksiköt ovat:

- gramma ("g")
- kilogramma ("kg") = 1000.0 g
- naula ("lb") = 453.6 g
- unssi ("oz") = 28.4 g
- millilitra ("ml")
- desilitra ("dl") = 100.0 ml
- litra ("l") = 1000.0 ml
- teelusikka ("tl") = 5.0 ml
- ruokalusikka ("rkl") = 15.0 ml
- kuppi ("cup") = 240 ml
- pinta ("pint") = 470 ml

Tiheys, erityisesti muodossa g/ml, on melko hankala käsittää intuitiivisesti, kun on luomassa aineita. Sen vuoksi ohjelmaan on lisätty tiheyslaskuri. Tiheyslaskuri pyytää käyttäjältä syötteenä massan arvon ja mittayksikön sekä tilavuuden arvon ja mittayksikön, ja laskee näiden suhteen avulla käyttäjälle tiheyden.

## Ohjelman rakenne

Ohjelma on jaettu kahdeksaan pääosaan: Aine-luokkaan, Hakukone-, Muuntaja-, Varasto-, IO-, UI- ja GUI-yksittäisolioihin sekä Poikkeukset poikkeusluokkiin. Näistä keskeisimmät ovat Aine, Varasto ja Hakukone; muut luokat lähinnä helpottavat näiden toimintaa.

Aine-luokka mallintaa reseptejä, ainesosia ja raaka-aineita, eli käytännössä kaikenlaisia aine-olioita. Aine-olioihin on tallennettu kaikki aineille ominainen tieto: nimi, allergeenit, tiheys, määrä (joka syntyy valmistettaessa), oletusmittayksikkö sekä ainesosat, eli mitä aineksia vaaditaan aineen valmistukseen. Lopuksi on myös aineen kuvaus, joka on yleisimmiten valmistusohje aineelle.

Varasto-olioon tallennetaan tiedot kaikista ohjelman tuntemista Aine-olioista, ja tieto kuinka paljon käyttäjällä on niitä varastossa. Varastolla on metodeja, joilla voidaan hallinoida näitä määriä ja välittää näitä tietoja eteenpäin erityisesti Hakukoneelle.

Hakukone-olio on työkalu, joka helpottaa reseptien löytämistä Varaston tiedoista. Sen metodeja käytetään palauttamaan ainelistoja. Aineita voidaan suodattaa nimen ja allergeenien perusteella. Kriteeriksi voidaan myös määritellä, että aine pitää pystyä valmistamaan varastossa olevista aineksista, tai siten että puuttuu n aineista.

Muuntaja-oliota käytetään tekemään muunnoksia eri massojen ja tilavuuksien mittayksiköiden välillä. Hakukone käyttää Muuntajaa tarkistaessaan riittävätkö Varaston ainekset, koska Varastossa olevat aineet

voivat olla eri mittayksikössä kuin reseptissä jota Hakukone tarkistaa.

IO-olio käsittelee ohjelman tietojen tallennustarpeita. Se tallentaa Aine-olioita omille tekstitiedostoilleen ja Varaston tiedot yhdelle tekstitiedostolle. Se voi myös välittää tiedot Varastolle tältä tiedostolta, ja luoda Aine-tiedostoilta uusia Aine-olioita. IO suorittaa näitä toimenpiteitä, kun ohjelma käynnistetään tai suljetaan, tai kun luodaan uusia aineita käyttöliittymän kautta.

GUI-olio on ohjelman graafinen käyttöliittymä, joka kutsuu pitkälti UI-olion metodeja. Sen kautta voidaan käyttää kaikkia ohjelman toimintoja, ja huomattavasti kätevämmän kuin merkkipohjaisesti. Käyttöliittymässä on viisi suurempaa näkymää, muut osat ovat lähinnä ponnahdusikkunoita ynnä muita vastaavia. Pääikkunassa on listattuna taulukkoon kaikki Varaston tuntemat aineet ja niiden määrät, sekä napit joista pääsee kolmeen muuhun ikkunaan. Reseptin luonti-ikkunassa voidaan interaktiivisesti luoda uusia Aine-olioita, ja muuttaa olemassa olevia. Reseptihaussa voidaan käyttää Hakukonetta aineiden etsimiseen. Varastonhallinnassa voidaan kätevästi muuttaa aineiden määriä Varastossa. Viimeinen ikkuna on Aineikkuna, jonka voi avata painamalla listattuja aineita eri näkymissä. Siinä voi tarkastella ja halutessa muuttaa aineen ominaisuuksia.

## Algoritmit

Valtaosa Muuntaja-olion metodeista toimii käyttäen kaavaa  $\text{tiheys} = \text{massa} / \text{tilavuus}$  ( $d = m / V$ ). Tätä kaavaa hyödyntääkseen Muuntaja laskee eri mittayksiköiden suhteet (esim.  $\text{kg/g} = 1000$ ), josta voi laskea muunnoksia, kun on kyse muunnos massayksiköstä massayksikköön tai vastaava tilavuuksien välillä. Tilavuuksien ja massojen välisiä muunnoksia varten yksiköt muutetaan vastaamaan Aine-olioiden käyttämää tiheyden yksikköä ( $\text{g/ml}$ ).

Aine-luokka ja Hakukone-olio käyttävät rekursiota käydessään läpi Aine-olioiden ainesosia. Metodeita Aine.aineetYhteensä, Aine.sisältääAineen ja Hakukone.voiValmistaa kutsutaan siis ensin kyseessä olevalle Aine-oliolle, ja sitten sen mahdollisille ainesosille ja niin edelleen, kunnes saavutetaan aineen raaka-aineet.

Hyvin monet metodit käyttävät silmukoita käymään läpi kokoelmia.

## Tietorakenteet

Varasto käyttää hakurakennetta (Map), koska se koostuu käytännössä täysin avain-arvo -pareista (aine-määrä). Muut tietorakenteet ovat lähinnä muuttuvatilaisia, koska halusin toteuttaa aineet siten, että niiden ominaisuuksia voi halutessa muuttaa tai korjata.

## Tiedostot

Ohjelma käsittelee IO-olion kautta kahdenlaisia tekstitiedostoja: Aine- ja Varasto-tiedostoja. Aine-oliot tallennetaan tekstitiedostoille, Reseptikirjan reseptikansioon. Sieltä ne voidaan lukea myöhemmin tarvittaessa. Tiedoston ensimmäiselle riville tulee aineen nimi. Toiselle riville tulee aineen tiheys, määrä ja mittayksikkö. Seuraaville riveille tulevat ainesosat, jokainen omalle rivilleen. Ainesosien jälkeen seuraavalla rivillä on tähtimerkki ("\*") erottimena, jonka jälkeen seuraavalla rivillä on aineen allergeenit. Viimeisillä riveillä on aineen kuvaus. Esimerkki:

*Spagetti bolognese*

*0.0,4.0,kpl*

*spagetti,300.0,g*

*kastike,800.0,g*

*\**

*liha,tomaatti*

*Spagetti bolognese, neljä annosta. Paista jauheliha... jne.*

Varaston tiedot tallennetaan riveittäin muodossa nimi ja määrä. Esimerkki:

*spagetti,300.0*  
*maito,1.0*  
*kananmuna,12.0*

## Testaus

Ohjelman testaus oli hyvin puutteellista. Tämä johtui siitä, että en saanut scalatest-kirjastoa, jota olin suunnitellut käyttäväni yksikkötestaukseen, toimimaan projektissani kuin vasta toisen checkpointin jälkeen. Testaus oli hyvin pitkälti scala-tulkilla ja silmämääräisesti tarkistaen, että esimerkiksi tekstitiedostot luotiin oikein ja pystyttiin myös lukemaan. Ajan loputtua kesken, ehdin toteuttaa suunnittelemani yksikkötesteistä ainoastaan metodin aineetYhteensä toimintaa testaavia.

Ohjelmaan tuli enemmän metodeja kuin olin alun perin suunnitellut, joten alkuperäinen testaussuunnitelmani ei kattanut aivan koko ohjelmaa. Valitettavasti en ehtinyt toteuttaa edes kaikkea suunnittelemani testausta.

Käyttöliittymän valmistuttua ohjelman toiminnot on testattu käsin. Ohjelman pitäisi täyttää kaikki tehtävänannon vaatimukset, vaikkakin on hyvin mahdollista, että on tapauksia, joita en ole ajatellut testata.

## Ohjelman tunnetut puutteet ja viat

Hakukoneen metodi voiValmistaa, jolla tarkistetaan voiko annetun aineen valmistaa varastossa olevilla aineilla, on puutteellinen. Tällä hetkellä se palauttaa oikean arvon, kun kyseessä on aine, jolla ei ole ainesosia, tai sen ainesosat ovat raaka-aineita (eli ainesosilla ei ole omia ainesosia). Väärä tulos saattaa tulla, kun esimerkiksi reseptiin A tarvitaan 2 ainetta B, jonka valmistamiseen puolestaan tarvitaan 4x C. Tämä virhe syntyy, koska metodi ei tee tarkistusta paljonko ainetta B syntyy kun sitä valmistetaan aineesta C. En keksinyt ratkaisua tähän ongelmaan.

Jos menee omakätisesti muuttamaan varasto-tekstitiedostoa, ja lisää sinne aineita, joita ei ole reseptikansiossa, Varaston tietoja ei pysty täyttämään kokonaan metodilla lataa(), koska syntyy poikkeus, kun yritetään lisätä ohjelmalle tuntematon aine.

Samanlaisia virheitä pystyy myös ehkä aiheuttamaan, jos lisää aineelle ainesosia, ja poistaa ne kesken ajon. Ohjelma poistaa ainesosaluettelosta puuttuvat ainesosat yleensä vain ohjelmaa käynnistettäessä.

Ohjelmassa ei ole ehkä tarpeeksi lisämäärittelyitä, kuten private, varsinkin koska suurinta osaa muuttujista voi joka tapauksessa muuttaa jollakin saman luokan metodilla.

En huomannut muita vikoja, mutta niitä on voinut päästä tarkistusteni ohi, koska en tehnyt kattavaa yksikkötestausta.

## Kolme parasta ja kolme heikointa kohtaa

Mielestäni paras kohta ohjelmasta on tekstitiedostojen asettelu. Aine-tiedostoja pystyy helposti kirjoittamaan suoraan tekstitiedostolle, ja niitä on helppo lukea, kun tunnistaa muuttujien järjestyksen. Tämä oli hyödyllistä testatessa ohjelmaa ilman käyttöliittymää, sillä se oli huomattavasti nopeampaa kuin kirjoittaa konstruktoreita yhteen pötköön Aine-olioille.

Toinen vahva kohta on mielestäni koodin luettavuus. En ole kokenut koodari, joten arvioini voi olla hyvinkin pielessä, mutta kehitystä on tapahtunut verrattuna vanhempiin ohjelmiini. Mielestäni ohjelma on suurimmaksi osaksi selkeä lukea ja ymmärtää mitä eri metodeissa tapahtuu. Loppupäässä toteutetut kohdat on toteutettu kiireellä, joten niissä voi olla puutteellisia kohtia. En kommentoinut kaikkea täysin tyhjentävästi, mutta mielestäni nimesin metodit ja niiden konstruktorit selkeästi. Se on siis ainakin

suhteellinen vahva kohta, jos ei absoluuttinen.

Kolmas vahva kohta on graafinen käyttöliittymä. Sitä lienee paikoittain hankala käyttää ilman dokumentin käyttöohjetta ja se on hyvin ruma, mutta se on huomattavasti kätevämpi kuin merkkipohjainen käyttöliittymä. Ajan kanssa olisin halunnut korvata monet tekstisyötteenä annettavat toiminnot, mutta nekin ovat nopeita, kun on muutaman kerran käyttänyt niitä.

Ensimmäinen heikkous on Hakukone. Se täyttää suurimman osan toiminnallisuudesta, mutta niin kuin mainittu viat ja puutteet -kohdassa, siinä on tapauksia, jossa se ei toimi täysin oikein. Puutteet eivät välttämättä ilmaannu arkisemmissa ruokalajeissa, mutta jos lähdetään lisäämään moniasteisia reseptejä, virheellisten hakutulosten riski kasvaa.

Toinen heikkous on että käyttöliittymä on epäoptimaalinen ja vaatii paljon merkkijonona annettavia syötteitä, joiden tulee olla melko tarkkaan oikein annettuja, muuten ne eivät toimi. Olen pyrkinyt selittämään syöteformaatin jokaisessa kohdassa, jossa sitä pyydetään, mutta koska en ole testauttanut ohjelmaa muilla ihmisillä, siinä saattaa olla hankaluuksia, joille olen itse sokea.

Kolmas heikkous on Aine-olioiden ja Varaston toiminta tekstitiedostojen kanssa. Tekstitiedostoja on helppo korruptoida, ja kuten aiemmin mainittu tämä hankaloittaa eritoten Varaston luomista. Aine-oliot ovat itsenäisiä, mutta jos yritetään löytää Varastolle olemattomia aineita, toiminta hankaloituu.

## Poikkeamat suunnitelmasta ja aikataulu

Alun perin suunnittelin, että ainesosat-muuttuja on Aine-luokan konstruktori, mutta siirsin sen tavalliseksi muuttujaksi. Tämä oli sen vuoksi, että Aine-olioiden luominen on huomattavasti hankalampaa, jos ohjelman täytyy tietää aineen kaikki ainesosat ennen luomista. On helpompaa luoda aine ensin, ja luoda ainesosat halutessa, ja sitten lisätä ne reseptiin.

Päädyin assistentin neuvosta myös jakamaan IO-yksittäisolion erilleen, mikä oli hyvä idea. Alunperin en ajatellut toteuttavani UI-oliota, mutta se olisi ollut virhe, ja tarpeettomasti monimutkaistanut GUI-olion koodia.

Ajankäyttösuunnitelmani ei osunut ollenkaan kohdilleen. Minulla meni Aine-, Varasto- ja Hakukone-luokissa luultavasti noin puolet siitä ajasta, jonka arvioin. Sen sijaan Muuntaja-luokassa kesti muutama tunti pitempään kuin suunnittelin. Graafisessa käyttöliittymässä meni merkittävästi kauemmin kuin odotin. Yritin toteuttaa liian monimutkaisia ominaisuuksia Swingillä aikatauluuni ja taitoihini nähden. Minulla tuli eteen ongelmia, jotka pysäyttivät täysin työskentelyni etenkin käyttöliittymän kanssa. Sain toteutettua muut luokat sen verran hyvässä tahdissa, etten varannut tarpeeksi aikaa käyttöliittymää varten.

Lähdin työstämään projektia liian myöhään, joka kostautui, kun eteen ilmestyi odottamattomia haasteita, joista suurin osa oli itse aiheutettuja. Työjärjestykseni oli suurin piirtein sama kuin suunnittelin, poikkeuksena Muuntaja, jonka toteutin ennen kuin viimeistelin Aine-luokan.

Aikataulut oli siis erittäin huono, ja venyi pahasti yliajalle.

## Arvio lopputuloksesta

Mielestäni ohjelmani on tyydyttävä. Sen pitäisi toimia kaikissa tilanteissa paitsi poikkeustilanteissa, joita en ole tajunnut kokeilla. Hakukoneessa on lieviä puutteita ja Varasto on jokseenkin korruptioherkkä, mutta ongelmien ei pitäisi ilmetä, jos ohjelmaa käyttää ohjeistuksen mukaisesti. Käyttöliittymä on kätevä, vaikkakin kenties hieman yksinkertainen. Uskoakseni mallinnukseni ja tietorakennevalinnat sopivat ohjelman toteutukseen. Toteutin kattavasti metodeja kaikille luokille,

joten erityisesti käyttöliittymää lienee helppo laajentaa suunnittelemastani. Jos päänsisäinen reseptikirjani koostuisi yli kymmenestä reseptistä, käyttäisin mielelläni ohjelmaa arjessa.

## **Viitteet**

Ohjelmointi 1 -kurssimateriaali

Ohjelmointistudio 2 -kurssimateriaali

Ohjelmointi 2 -kurssimateriaali

scala API

scala.swing API

stackexchange.com

tutorialspoint.com/scala

## **Liitteet**

Smart\_Cookbook -Scala-projekti